Final Report

CSE-0302 Summer - 2021

Md.Khaled Rahman [UG02-50-19-003]
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
Email: muhammedkhalid70@gmail.com

***Abstract*—Main theme of your assignment or academic projects.**

n

***Index Terms*—The word mostly used in your report.**

I. Introduction

Assignment 4 : Detecting Simple Syntax Errors

Syntax errors are very common in source program.The main purpose of this session is to write programs to detect and report simple syntax errors.

Assignment 5 : Use of CFGs for Parsing

We can think of using CFGs to parse various language constructs in the token streams freed from simple syntactic and semantic errors, as it is easier to describe the constructs with CFGs.But CFGs are hard to apply practically. In this session,we implement a simple recursive descent parser to parse a number of types of statements after exercising with simpler CFGs.We note that a recursive decent parser can be construsted from a CFGs with reduced left recursion and ambiguity.

Assignment 6 : Predictive Parsing

Manual implementation of LL(1) and LR(1) parsing algorithms .

II. Literature Review

Assignment 4 : Detecting Simple Syntax Errors

A frustrating aspect of software development is that compiler error messages often fail to locate the actual cause of a syntax error.Syntax Errors Just Aren't Natural.Jashua Charles (Department of Computing Science),Abram Hindle (department of Computing Science),Jose Nelson Amaral(Department of Computing Science) Improving Error Reporting with Language Models.

Assignment 5 : Use of CFGs for Parsing

Context Free Grammars (CFG) can be classified on the basis of following two properties: 1) Based on number of strings it generates. During Compilation, the parser uses the grammar of the language to make a parse tree(or derivation tree) out of the source code. Vilhjálmur orsteinsson, Hulda Óladóttir,Hrafn Loftsson(Department of Computer Science) .Both present open-source,wide-coverage context-free grammer (CFG) for Icelandic and an accompanying parsing system.

Assignment 6 : Predictive Parsing

A predictive parser is a recursive descent parser with no backtracking or backup. It is a top-down parser that does not require backtracking. At each step, the choice of the rule to be expanded is made upon the next terminal symbol.

III. Proposed Methodology

IV. Conclusion and Future Work

Every Computer Engineer should learn compiler design so that an interpreted scripting language and interpreter.I think thatwhat is useful is how to :Parse an expression tree,Robust error handling,General-purpose text processing technique,Sanitize input,Schedule tasks in the future with cross-platform timers,Creation of virtual machines.

Acknowledgment

I would like to thank my honourable **Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

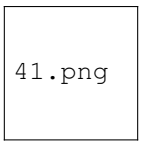

Fig. 1. Proposed Methodology

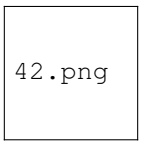

Fig. 2. Proposed Methodology

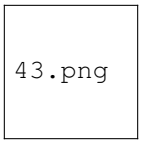

Fig. 3. Proposed Methodology

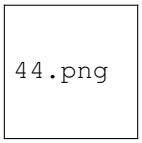

Fig. 4. Proposed Methodology

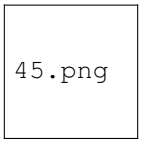

Fig. 5. Proposed Methodology

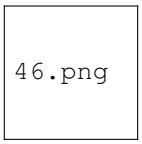

Fig. 6. Proposed Methodology

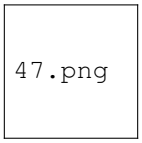

Fig. 7. Proposed Methodology

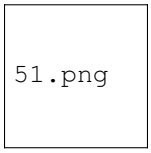

Fig. 8. Proposed Methodology

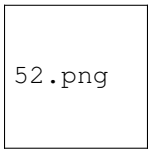

Fig. 9. Proposed Methodology

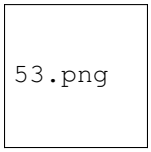

Fig. 10. Proposed Methodology

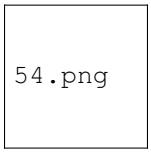

Fig. 11. Proposed Methodology

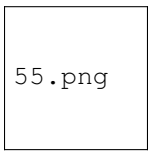

Fig. 12. Proposed Methodology

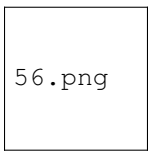

Fig. 13. Proposed Methodology

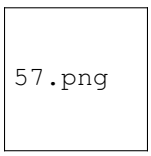

Fig. 14. Proposed Methodology

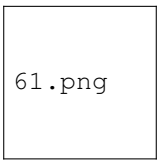

Fig. 15. Proposed Methodology

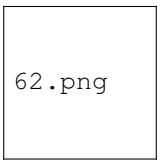

Fig. 16. Proposed Methodology

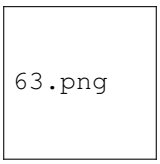

Fig. 17. Proposed Methodology

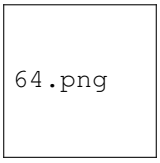

Fig. 18. Proposed Methodology

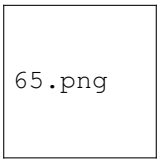

Fig. 19. Proposed Methodology

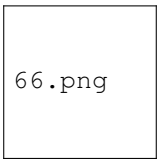

Fig. 20. Proposed Methodology

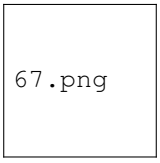

Fig. 21. Proposed Methodology

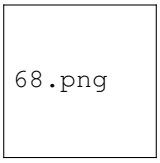

Fig. 22. Proposed Methodology

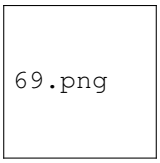

Fig. 23. Proposed Methodology

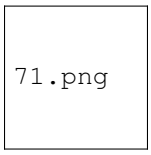

Fig. 24. Proposed Methodology

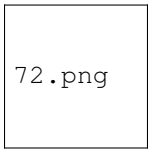

Fig. 25. Proposed Methodology

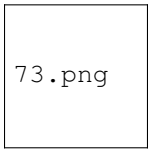

Fig. 26. Proposed Methodology

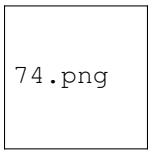

Fig. 27. Proposed Methodology

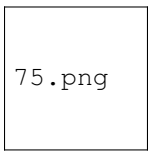

Fig. 28. Proposed Methodology